ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022
Paper ID #37356

Work in Progress: A Literature Review On Computational & Numerical Methods in Engineering Education

Joseph Marie Nicolas Léger

Nicolas Léger is currently an engineering and computing education Ph.D. student in the School of Universal Computing, Construction, and Engineering Education (SUCCEED) at Florida International University. He earned a B.S. in Chemical and Biomolecular Engineering from the University of Maryland at College Park in May 2021 and began his Ph.D. studies the following fall semester. His research interests center on numerical and computational methods in STEM education and in Engineering Entrepreneurship.

Bruk T Berhane (Assistant Professor of Engineering Education)

A Literature Review On Computational & Numerical Methods in Engineering Education (Work in Progress)

Abstract

Scholars within computing and engineering education have broadly explored how students understand computing-related disciplines. Existing studies have led to publications on promising practices for teaching computing, such as evidence-based strategies for teaching computer science principles and programming languages like C++ and Java. However, other education research on computer-based technologies that students in engineering disciplines often use, such as MATLAB, Mathematica, Mathcad, Excel, and Aspen Plus, is limited. Initial analysis reveals that most of the published work on numerical and computational methods (NCM) in engineering education is in the form of textbooks used to introduce these software packages. However, a gap in the literature exists between understanding the technical content that these textbooks provide, and the degree to which effective pedagogy is used to teach this content. In other words, while textbooks exist to introduce these concepts, few studies have been done to measure the effectiveness of teaching these concepts. Understanding how to effectively teach these computing resources is important because it provides a foundation to allow engineering students to tackle their day-to-day calculations and be prepared with essential tools in the 21st century to tackle modern projects. Literature indicates that very little has been done to understand the impact of these computational technologies both on students as well as the perspective of faculty who teach these courses. As part of this literature review, we will review published scholarship to understand ways in which scholars have investigated the way that computational and numerical methods have been taught so far. This is a work in progress that will serve as a foundation for a broader study to understand effective pedagogies in computational and numerical methods in computing and engineering education.

Keywords: Numerical Methods, Computational Methods, Mathematics, Undergraduate Engineering Education, STEM Computational Methods
Paper type: Literature Review

Introduction & Background

With the advancement of technology, more and more computer-based tools have been created over time and implemented in the world of engineering and allowed to perform more complex calculations and solve problems with greater accuracy. Consequently, machines have been able to help humans with their computational needs. Examples include the Leibniz machine back in 1673. In that regard, Leibniz, who was a prominent philosopher and mathematician stipulated that "[…] it is unworthy of excellent men to lose hours like slaves in the labor of calculation which could be safely relegated to anyone else if machines were used [1, vol.1, p. 651]." The earlier computing tools include the log scale table, created by John Napier around 1594 [2], the slide rule by William Oughtred in 1622 [3], mainframe computers like the IBM Card Programmed Calculator in 1948 among other computing devices in that era. With the advent of transistors in the 1970s, Hewlett-Packard created the first programmable scientific calculators at a significantly lower cost [4]. By the mid-1990s, the personal computer (PC) became an essential component of most college students' education [5] due to the significant decrease in size, cost, and electrical consumption of electronics, namely, transistors.

Fast forward to the 21st century, these nano-transistors [6] are powering PCs that run software where users can input commands in order to compute specific tasks.

Namely, the following software types and software are currently used in Engineering:

- Programming languages such as Fortran and Python
- Spreadsheets such as Microsoft Excel, Google Sheets, and Origin Pro
- Software packages such as Mathcad, MATLAB, Maple, and Matematica
- Process simulators such as DWSIM and Aspen Plus
- Computer-aided design such as AutoCAD and Creo Parametric

Positionality Statement

In this paper, the primary author draws from his experiences using computational tools as an undergraduate student in chemical engineering. As an undergraduate student, he along with several fellow students struggled with a lack of understanding in a required course in numerical and computational methods (NCM). Despite the fact that he obtained a good grade in this course, he did not retain or fully grasp much of the information taught in this course. He eventually learned computational methods while working at research laboratories and by doing some personal research on mathematical modeling. Thus, he is motivated to explore the literature on the topic of numerical and computational methods in engineering and computing education, which can subsequently help to broaden the understanding of these methods and ensure that future students have a better experience learning and applying these tools in their work.

Literature Review Methodology

Based on the primary author's curiosity on the subject of numerical and computational methods in undergraduate engineering education, he distilled this question: *What challenges and opportunities have been identified in existing scholarship for advancing teaching and learning numerical and computational methods in undergraduate engineering education?* From that question, the following keywords were utilized for our initial search: "students' learning / computational OR numerical methods/job OR profession OR career/ education OR college or university." Then the keywords were entered in databases such as Compendex (Engineering Village), ProQuest (Elsevier), and ACM Digital Library. Based on this search process, we mainly identified textbooks that were used to introduce these software packages. These textbooks covered topics such as matrices, linear regression, and differential equations among other numerical methods. However, results from this initial search did not provide sufficient information to answer the research question. The primary author aimed to find results related to pedagogical scholarship or epistemology behind specific teaching methods. We then modified our results to include the following search terms: "teach numerical methods AND MATLAB or Excel, undergraduate engineering AND programming," which yielded some interesting results both on Microsoft Academic and Google Scholar. To that end, this paper presents a work-in-progress literature review that uses the terms from this modified search. Results were obtained from both Microsoft Academic and Google Scholar. A systematic review will be conducted in future versions of this work.

Time Period for Citations Included in this Literature Review

This work-in-progress explores the literature on the topic of numerical and computational methods in undergraduate engineering education and on how computational technologies are being implemented so far in the 21st-century engineering classroom. In their paper entitled "*Five Major Shifts in 100 Years of Engineering Education*", Froyd et al., stated that the fifth shift in engineering education is the influence of information, communication, and computational technologies (ICCT) that has evolved considerably over the last two decades [7]. The literature reviewed in this paper, therefore, covers the time period from 2000 to the present. It is also worth noting that a desktop version of MATLAB, used by many engineering professors, was introduced in 2000 [8]. Other computing packages, such as Mathcad, Aspen Plus, Maple, Matematica, and Microsoft Excel also grew in popularity around that time [9]-[11], [24], [27].

Tools Showing Promise Across Engineering Disciplines

As part of this fifth major shift in engineering education, computational technologies have progressively replaced electronic calculators in the form of spreadsheets, software packages, and simulators, and consequently provided unmatched computational power for numerical analysis[7],[26]. While these technologies have been progressively implemented in the classroom, engineering instructors have adapted them in different ways in their classrooms. Below we offer a short overview of studies, selected to show different engineering disciplines and the use of computational technologies in their respective fields. We do not claim that the studies within these specific disciplines are generalizable across each entire field but provide this summary as evidence of some promising approaches that have been noted in different engineering fields. In the finalized version of this manuscript, we will include additional publications, including a number of citations that indicate how NCM has been used for teaching engineering.

Rao et al., in electrical engineering, as part of a power electronics (PE) course, found that the use of MATLAB/Simulink is highly effective at simulating PE circuits, by allowing real-time simulations and interactions with the material [12]. The MATLAB/Simulink environment has the added benefit of being inexpensive [7] with no need for “real hardware” [12], thus making this technology more accessible to engineering students and faculty. Similarly, Guzmán et al., in chemical engineering, coded an “interactive software tool developed in support of system identification education” using Sysquake, a programming language similar to MATLAB, and found it “very useful from an educational point of view [13].” Overall, both Rao et al. [12] and Guzmán et al. [13] found that these interactive tools enable the students to gain a better understanding of the fundamentals in their respective courses compared to other existing ways.

On the other hand, Fernández et al. [15], in marine engineering, found that their students spend too much time learning to program in their classes instead of the engineering methods and that graduates of the program tend to reject “all the subjects related with these applications, thus limiting their working abilities and expectations when they complete their studies [15].” In other words, they observed that their students tend to become mired in learning how to program, leading to them avoiding using programming for their engineering tasks, thus impeding their work both in their classes and in the field.

Consequently, after analyzing mathematical computing software like “C, Matlab, Mathematica, Pascal, FORTRAN, Visual Basic, and VBA”, they found Microsoft Excel more user friendly and allows for a better understanding of “the theory employed to solve each equation and to differentiate between each of the numerical methods in parameters like time and number of iterations[15, p.29].” However, in their research, they found a scarcity of pre-existing resources on numerical methods using Microsoft Excel. Consequently, Fernández and colleagues created their own approach to using Microsoft Excel in order to facilitate their students' learning [15].

They furthermore suggested that the use of Visual Basic with Microsoft Excel could provide a stronger grounding for teaching numerical methods [15].

Similarly, Demir et al. [22], in environmental engineering, in the context of a water distribution networks (WDN) course used Microsoft Excel with Visual Basic. In terms of accessibility, they suggested that Microsoft Excel with Visual Basic comes in most personal computers and thus, is more accessible compared to MATLAB. They also acknowledged that MATLAB "could provide much faster solutions than Excel VBA[22]." They also found several promising approaches for the use of Microsoft Excel, including the Hardy-Cross method and Newton–Raphson method [22].

More recently, in control engineering, Matlab live scripts (.mlx) have been found to provide a more interactive "computer-based control" learning environment compared to traditional (.m) files that only offer a coding environment [25]. In addition, Matlab live scripts have the added benefit of combining code, text, equations, and lively generated graphs and tabulated outputs and thus, allows for a richer learning experience. Consequently, Nevaranta et al. [25] recommend the use of live scripts in other branches of engineering.

Stephen J. Eglen [26], in computational biology, uses R as part of his class. Eglen had to create extra resources for his master's students due to their varying backgrounds in programming experience. R in the field of computational biology is one of the preferred languages due to the existence of "set packages to analyze genomic data [16]." Similarly to the other computational methods discussed so far, R is capable of outputting some "high-quality" graphics and allows for a modeling and analysis environment [16].

Again, these initial studies are included because they illustrate the breadth of scholarship, albeit limited, across different engineering fields. In future iterations, we can include more citations that focus on engineering pedagogy using NCM. We also will, where possible, include quantitative data from prior studies that indicate, for example, the number of students impacted by a particular NCM intervention; we note that thus far, our searches have revealed little quantitative data from prior studies. Instead, these prior works focus on overarching themes related to NCM across engineering disciplines.

Lack of Programming Skills Among College Students and Faculty Preparation

Several studies suggest that undergraduate [15],[19] and graduate [16] students lack programming skills coming in the classes that require them. As an example, both Fernández et al. [15] and Eglen [16] innovated and created new teaching and learning approaches to help students with weaker backgrounds learn to perform computational calculations. To ensure that students are better prepared by the time they reach post-secondary education, Barr et al. [17] suggest implementing "computational thinking concepts into the K-12 curriculum[17]." They also recommend new educational policies and better training and resources for K-12 educators.

To date, computing-related education work in K-12 such as Computer Science For All [29], Black Girls Code [30], and Scratch [31] has elevated the relevance of programming. These initiatives have provided a platform and framework for children to learn about computer programming in an engaging way, thereby preparing them for college majors and future careers that require these skills. Indeed, the movement towards more computational thinking [17] in high school also undergirds the relevance of these programming efforts. At the same time, scholarship reveals almost no evidence of implementing computational tools such as MATLAB or Mathematica, in high schools. This suggests an opportunity for computational tools to be included within the broader discourse of K-12 computing education. Future work includes an analysis of an NCM lesson plan embedded within a larger high school engineering curriculum. This type of research will underscore the importance of focusing on NCM in secondary school settings, thereby preparing high school students for post-secondary engineering education.

Conclusion

This work in progress paper provides an initial review on the subject of advancing the teaching and learning of numerical and computational methods in undergraduate engineering education. Initial findings were largely organized engineering disciplines and findings from scholars who have explored computational methods within these fields. In future manuscripts, we hope to further numerical and computational methods, looking at variations in engineering courses at different institutions to understand the extent to which pedagogical methods vary. We also hope to develop and implement a framework for teaching and learning these methods that are informed by promising practices from prior studies. Our long-term goal is to test the efficacy of this framework within multiple undergraduate engineering contexts. These may include both undergraduate classroom environments, as well as co-curricular spaces such as co-ops, which may concretize lessons learned in the school [18].

Our research findings in this paper also provide the option of transitioning into the K-12 engineering education space. Building on the work on computational thinking in the K-12 arena and scholars who have explored the utility of novel programming approaches, novel research on computational methods can complement this already growing body of work. To the extent that children can be exposed to programming languages like Java while also learning tools like MATLAB, they will be even more prepared for engaging careers in engineering. We hope to contribute to the production of future engineers who are academically and professionally prepared to solve challenges with the most advanced tools available.

References

[1]P. A. Laplante, Ed., Encyclopedia of Computer Science and Technology, 2nd ed. Boca Raton: CRC Press, 2017. doi: 10.1081/E-ECST.

[2]K. M. Clark and C. Montelle, "Logarithms: The Early History of a Familiar Function," The MAA Mathematical Sciences Digital Library, Washington, DC, Jun. 2010. doi: 10.4169/loci003495.

[3]H. Bruderer, "Slide Rules," in Milestones in Analog and Digital Computing, H. Bruderer, Ed. Cham: Springer International Publishing, 2020, pp. 543–592. doi: 10.1007/978-3-030-40974-6_14.

[4]"HP Virtual Museum: History of the 9100A desktop calculator, 1968." http://www.hp.com/hpinfo/abouthp/histnfacts/museum/personalsystems/0021/0021history.html (accessed Feb. 03, 2022).

[5]V. Utgikar, Fundamental concepts and computations in chemical engineering. Boston: Prentice Hall, 2017.

[6]"Introducing the world's first 2 nm node chip," IBM Research Blog, Feb. 09, 2021. https://research.ibm.com/blog/2-nm-chip (accessed Feb. 03, 2022).

[7]J. E. Froyd, P. C. Wankat, and K. A. Smith, "Five Major Shifts in 100 Years of Engineering Education," Proceedings of the IEEE, vol. 100, no. Special Centennial Issue, pp. 1344–1360, May 2012, doi: 10.1109/JPROC.2012.2190167.

[8]C. Moler and J. Little, "A history of MATLAB," Proc. ACM Program. Lang., vol. 4, no. HOPL, pp. 1–67, Jun. 2020, doi: 10.1145/3386331.

[9]J. N. Harb, A. Jones, R. L. Rowley, and W. V. Wilding, "Use of Computational Tools in Engineering Education: A Case Study on the Use of Mathcad," Chemical Engineering Education (CEE), vol. 31, no. 3, pp. 180–87, 1997.

[10]H. S. Fogler and N. M. Gurmen, "Aspen Plus™ Workshop for Reaction Engineering and Design," p. 44.

[11]N. Chonacky and D. Winch, "Maple, Mathematica, and Matlab: the 3M's without the tape," Computing in Science Engineering, vol. 7, no. 1, pp. 8–16, Jan. 2005, doi: 10.1109/MCSE.2005.18.

[12]P. V. V. R. Rao, G. Durga Prasad, and S. Dileep Kumar Varma, "Pedagogical Approach to Teach the Modeling of Power Electronic Converters," in 2014 IEEE Sixth International Conference on Technology for Education, Dec. 2014, pp. 191–192. doi: 10.1109/T4E.2014.59.

[13]J. Guzmán, D. Rivera, S. Dormido, and M. Berenguel, “Teaching System Identification Through Interactivity,” IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 8, Jan. 2009.

[14]Y. Dori, J. Belcher, M. Bessette, M. Danziger, A. McKinney, and E. Hult, “Technology for Active Learning,” Materials Today, vol. 6, pp. 44–49, Dec. 2003, doi: 10.1016/S1369-7021(03)01225-2.

[15]S. Fernández, J. A. Orosa, and J. J. Galán, “A New Methodology to Teach Numerical Methods With MS Excel,” Journal of Maritime Research, vol. 9, no. 2, Art. no. 2, 2012.

[16]S. J. Eglen, “A Quick Guide to Teaching R Programming to Computational Biology Students,” PLOS Computational Biology, vol. 5, no. 8, p. e1000482, Aug. 2009, doi: 10.1371/journal.pcbi.1000482.

[17]V. Barr and C. Stephenson, “Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?,” ACM Inroads, vol. 2, no. 1, pp. 48–54, Feb. 2011, doi: 10.1145/1929887.1929905.

[18]M. F. Cox, O. Cekic, and S. G. Adams, “Developing Leadership Skills of Undergraduate Engineering Students: Perspectives from Engineering Faculty,” Journal of STEM Education: Innovations and Research, vol. 11, no. 3, May 2010, Accessed: Jan. 29, 2022. [Online]. Available: https://www.jstem.org/jstem/index.php/JSTEM/article/view/1605

[19]S. Secules, A. Gupta, A. Elby, and C. Turpen, “Zooming Out from the Struggling Individual Student: An Account of the Cultural Construction of Engineering Ability in an Undergraduate Programming Class,” Journal of Engineering Education, vol. 107, no. 1, pp. 56–86, 2018, doi: 10.1002/jee.20191.

[20]N. A. M. Radzi et al., “Integrating programming with BeagleBone Black for undergraduate’s ‘programming for engineers’ syllabus,” in 2016 IEEE 8th International Conference on Engineering Education (ICEED), Dec. 2016, pp. 12–15. doi: 10.1109/ICEED.2016.7856055.

[21]P. Askar and D. Davenport, “AN INVESTIGATION OF FACTORS RELATED TO SELF-EFFICACY FOR JAVA PROGRAMMING AMONG ENGINEERING STUDENTS,” The Turkish Online Journal of Educational Technology, vol. 8, no. 1, p. 7, 2009.

[22]P. L. Li, A. J. Ko, and J. Zhu, “Appendix to: What Makes A Great Software Engineer?,” p. 76.

[23]S. Demir, N. Manav Demir, and A. Karadeniz, “An MS Excel tool for water distribution network design in environmental engineering education,” Computer Applications in Engineering Education, vol. 26, no. 2, pp. 203–214, 2018, doi: 10.1002/cae.21870.

[24]M. Niazkar and S. H. Afzali, “Application of Excel spreadsheet in engineering education,” p. 7, 2015.

[25]N. Nevaranta, P. Jaatinen, K. Gräsbeck, and O. Pyrhönen, “Interactive Learning Material for Control Engineering Education Using Matlab Live Scripts,” in 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Jul. 2019, vol. 1, pp. 1150–1154. doi: 10.1109/INDIN41052.2019.8972282.

[26]D. Ibrahim, “Engineering simulation with MATLAB: improving teaching and learning effectiveness,” Procedia Computer Science, vol. 3, pp. 853–858, 2011, doi: 10.1016/j.procs.2010.12.140.

[27]Baker, J. and Sugden, S., “Spreadsheets in education – The first 25 years,” Spreadsheets in Education, 1(1), article 2, 2003.

[28]*Engineering in K-12 Education: Understanding the Status and Improving the Prospects* (p. 12635). (2009). National Academies Press. https://doi.org/10.17226/12635

[29]A. Montoya, “Computer Science for All: Opportunities Through a Diverse Teaching Workforce,” *Harvard Journal of Hispanic Policy*, vol. 29, pp. 47–62, 2017.

[30]“Black Girls Code — Women of Color in Technology,” *Black Girls Code — Women of Color in Technology*. https://www.blackgirlscode.com.

[31]J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, “The Scratch Programming Language and Environment,” *ACM Trans. Comput. Educ.*, vol. 10, no. 4, pp. 1–15, Nov. 2010, doi: 10.1145/1868358.1868363.